

WSV303



web/services

Using XSD, CLR Types, And Serialization In Web Services

Doug Purdy
Program Manager
Microsoft Corporation
douglasp@microsoft.com

PDC⁰³

Make the connection

Microsoft[®]

WinFX Developer Preview

Tools

Visual Studio .net
Visual Basic .net
Visual C#.net
Visual J#.net
Visual C++.net

Client Application Model

Avalon

System.Windows

Windows Forms

System.Windows.Forms

Web & Service Application Model

ASP.NET / Indigo

System.Web

Data Systems Application Model

Win FS

System.Storage

Yukon

System.Data.SqlServer

Mobile PC & Devices Application Model

Compact Framework

System.Windows.Forms

Mobile PC Optimized

System.Windows

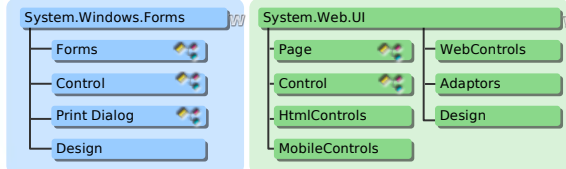
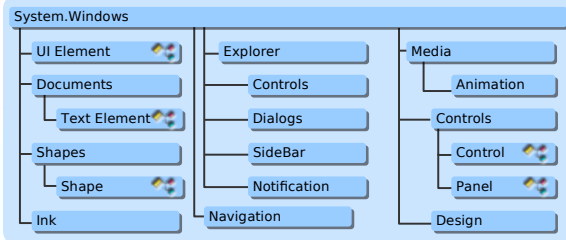
Command Line

System.Console

NT Service

System.ServiceProcess

Presentation



System.Help

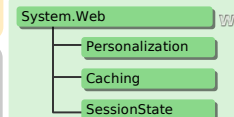
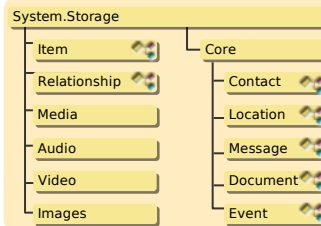
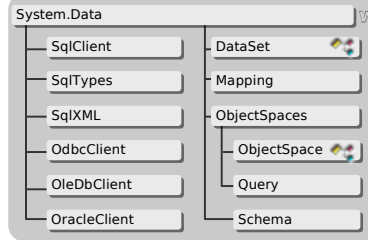
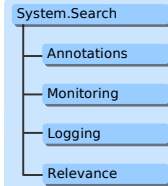
System.Drawing

System.NaturalLanguageServices

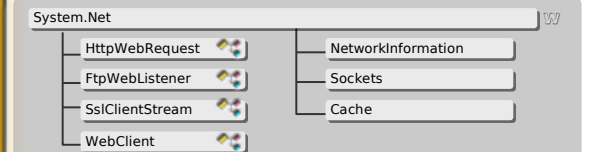
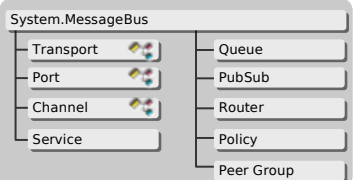
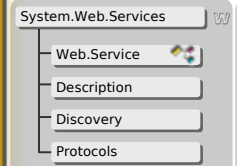
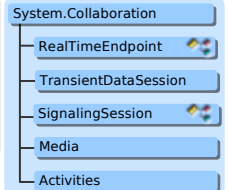
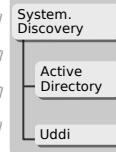
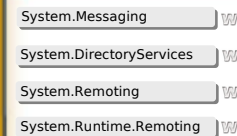
System.Speech

- Recognition
- Synthesis

Data

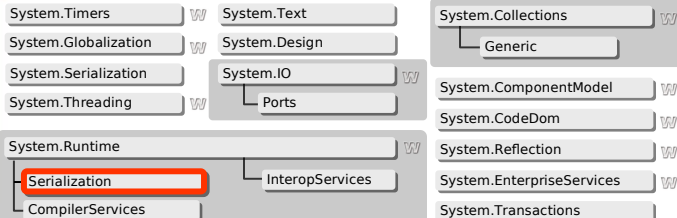


Communication

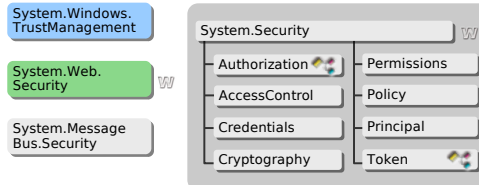


Fundamentals

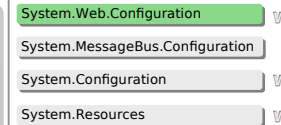
Base & Application Services



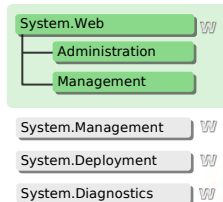
Security



Configuration



Deployment/Management



Agenda

- Overview
- What's new in "Whidbey"
- What's new in "Indigo"
- Moving to "Indigo"
- What's in the PDC bits
- Wrap Up
- Discussion

Serialization Overview

- The transformation of a graph of objects to a stream and back again
 - System.Xml.Serialization
 - System.Runtime.Serialization
- The transformation of a CLR type into an another type system (XSD) and back again
 - System.Xml.Serialization
- Framework-wide feature
 - Web Services, Remoting, Yukon, etc.
 - End-user feature

Serialization In “Whidbey”

What's New For Whidbey

- XmlSerializer

- New imperative model
- Schema import extensibility
- Serializer pregeneration
- Generics support
- SqlTypes

- BinaryFormatter

- Version tolerant serialization
- Performance improvements
- Generics support

Version Tolerant Serialization

demo

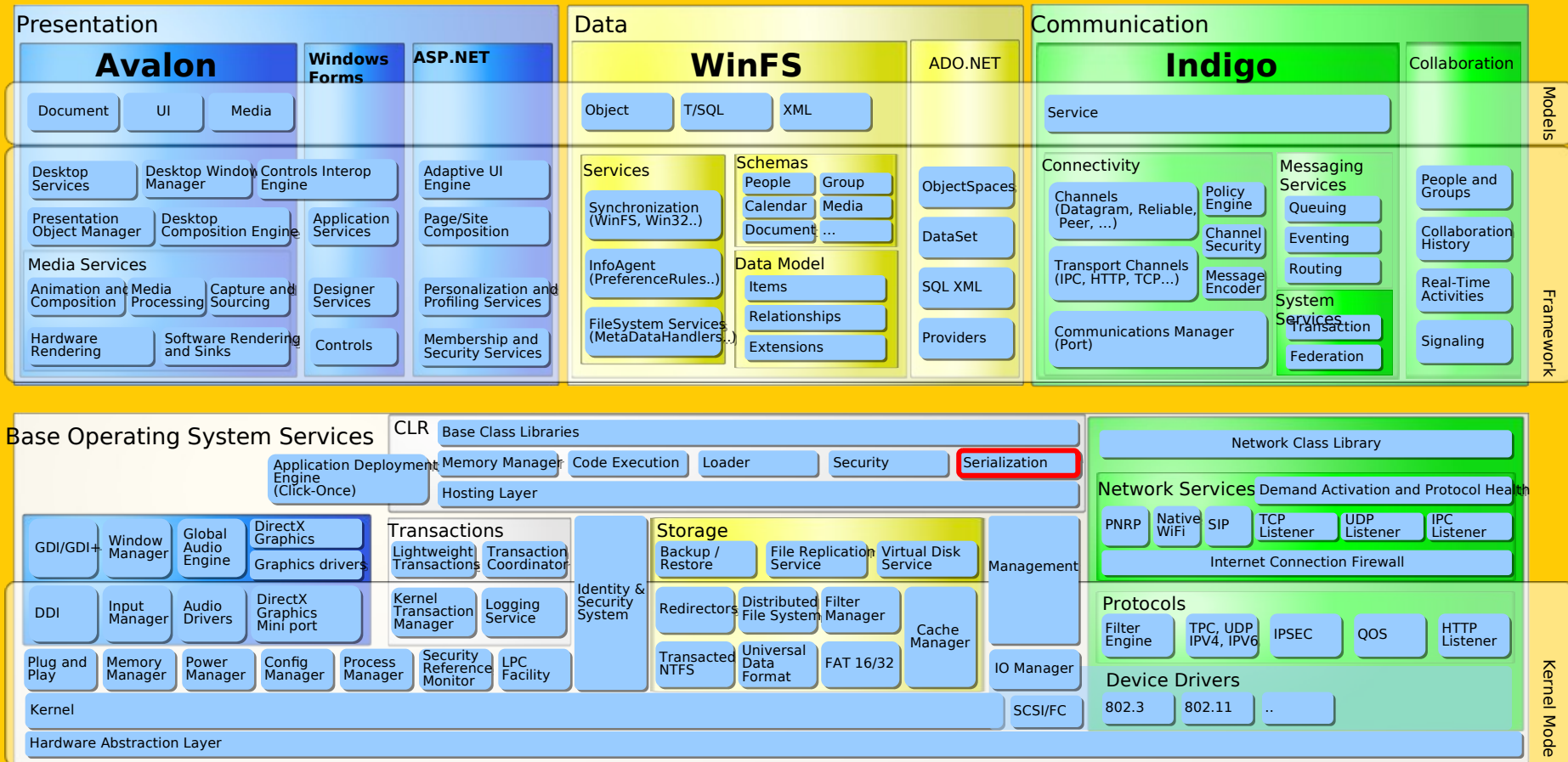
System.Runtime.Serialization

PDC⁰³

Make the connection

Serialization In “Indigo”

Longhorn Architecture



“Indigo” Architecture

PDC⁰³

Make the connection

Service Model

Instance Manager Context Manager Service Methods Type Integration Declarative Behaviors Transacted Methods

Connector

Channels
(Datagram, Reliable, Peer, ...)

Policy Engine

Transport Channels
(IPC, HTTP, TCP...)

Channel Security

Message Encoder

Communications Manager (Port)

Hosting Environments

ASP.NET .container.exe NT Service DllHost

Messaging

Queuing

Routing

Eventing

...

System Services

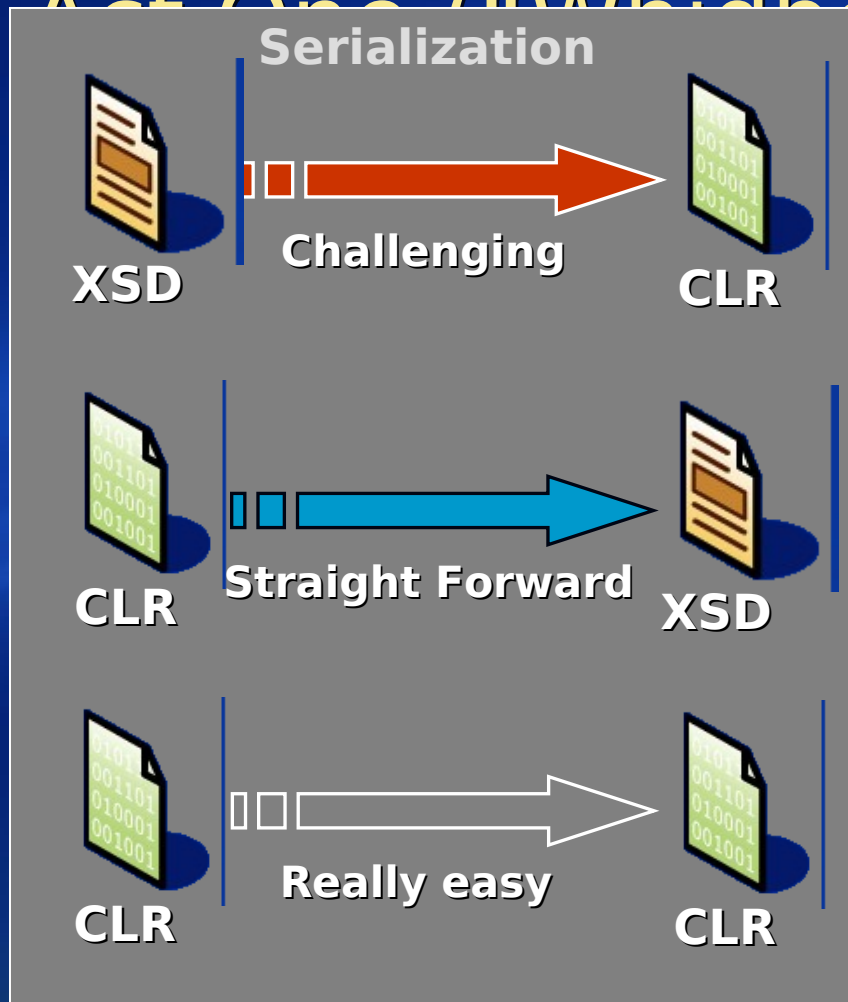
Transaction

Federation

...

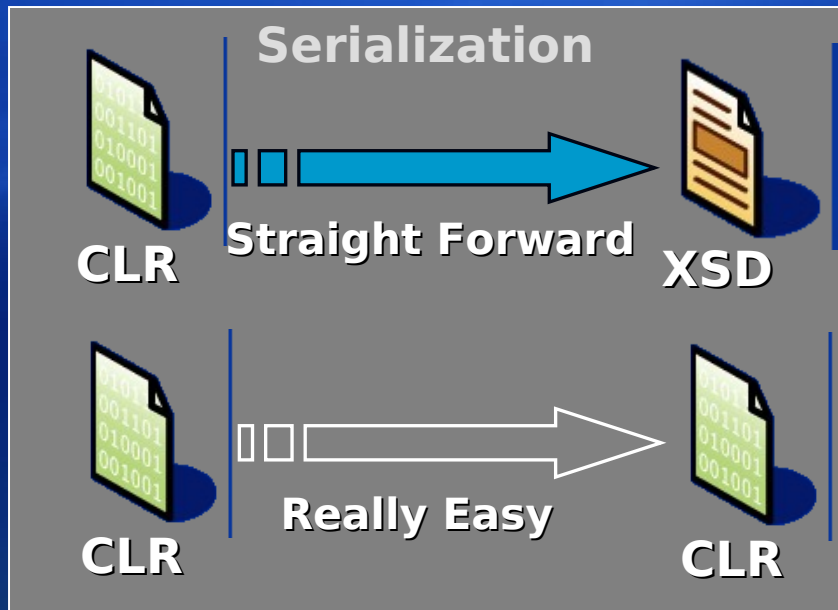
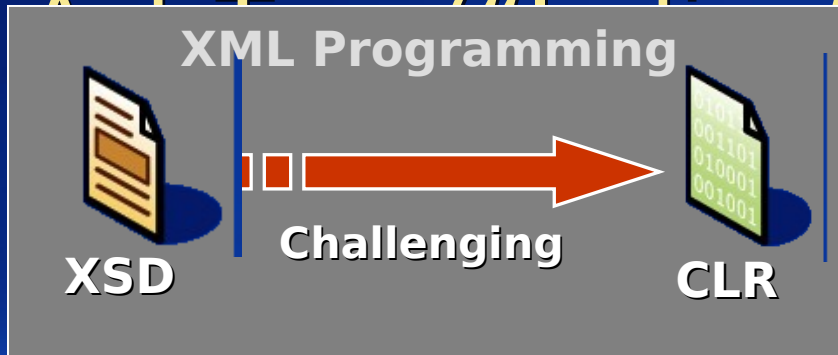
A Tale Of Two Type Systems

Act One ("Whodunnit")



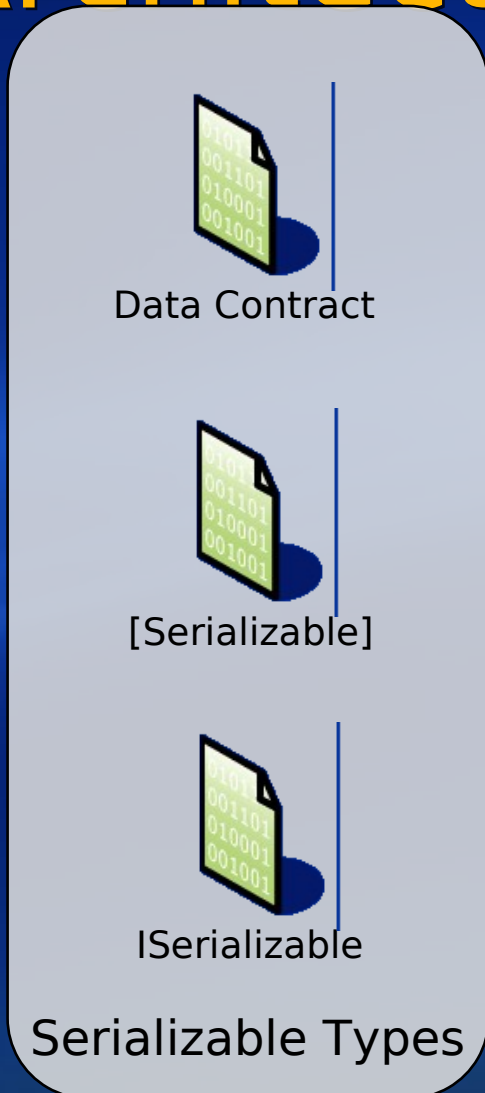
- XmlSerializer
 - Provides CLR programming model for some XSD constructs
 - Maps some CLR types into XSD
- SoapFormatter
 - Maps some CLR types (simple types) into XSD
 - Custom SOAP encoding for the rest
- BinaryFormatter
 - No support for XSD
 - CLR \leftrightarrow CLR only

A Tale Of Two Type Systems



- There are really two areas
 - Xml Programming
 - Serialization
- Xml Programming is about, well, Xml
 - XmlSerializer
 - XPathNavigator
 - ...
- Serialization is about making CLR types
 - Interoperable (XSD)
 - Loosely coupled
- Enter the XmlFormatter...

XmlFormatter Architecture



Serialization Engine

XmlFormatter
SerializationContext



Extensibility

Surrogates
Type Binders



Xml Processing

XmlReader
XmlWriter



text/xml



binary/xml

Representations

Serializable Types

- Existing Types
 - [Serializable] & ISerializable types just work
 - Gain loose coupling & version resilient
- Data Contract
 - New explicit programming model
 - Enables loose coupling for CLR types
 - Version resilient by default
 - Foundation for the future
- Working Together
 - Inheritance and composition just work

XmlFormatter And Serializable Types

demo

System.Runtime.Serialization

“Indigo” XSD Support

- Every serializable type is interoperable
 - We export valid XSD for any CLR construct
 - Our mapping is not extensible (POR for v1)
- Every XSD schema is supported
 - XmlFormatter
 - We provide a canonical CLR \leftrightarrow XSD mapping
 - No mapping extensibility (POR for v1)
 - XML programming models

XSD Support demo

System.Runtime.Serialization

Moving To “Indigo”

- “Indigo” works well with COM+, MSMQ, WSE, and ASMX apps
- “Indigo” nodes are easy to integrate into existing systems
- Existing applications have an easy migration path to “Indigo”
 - Existing serializable types and serialization extensibility points just work
- Follow the prescriptive guidance for writing distributed applications
 - Read and follow the serializable type guidelines

About The PDC Bits

- Two “Whidbey” features are available
 - `IXmlSerializable`
 - `SqlTypes`
 - The rest are being coded as we speak
- `System.Serialization`
 - Successor to the `XmlSerializer`
 - Yet another XML programming model
 - Taught us invaluable lessons
 - Will likely not ship with “Indigo”

For More Information

- Anytime

- E-Mail: douglasp@microsoft.com
- Blog: <http://www.douglasp.com>
- LHDC: <http://msdn.microsoft.com/longhorn>
- NS:
microsoft.public.windows.developer.winfx.indigo

- At PDC

- Hands on labs: On-site or download from CommNet
- Web/Services Lounge: 309 Foyer
- Ask The Experts: Tuesday 7 P.M. – 9 P.M. Hall G, H
- PDC Weblogs: <http://pdcbloggers.net>

This Talk In A Nutshell

- “Whidbey” offers a number evolutionary features to current serialization stacks
- “Indigo” delivers a unified serialization engine
 - Interoperable
 - Backward compatibility
 - Version resilient
 - High performance
- “Indigo” supports many XML programming models
- We need your feedback!!!

